

## APPENDIX B

### Class HttpURLConnection

Q B W P B \ 1 6 1 5 6 6 . 1

**Class Tree [Deprecated](#) [Index](#) [Help](#)**

[PREV CLASS](#) [NEXT CLASS](#)  
[SUMMARY](#): [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)  
[DETAIL](#): [FIELD](#) | [CONSTR](#) | [METHOD](#)

com.ibm.speech.net.www.protocol.https

**Class HttpURLConnection**

```

java.lang.Object
|
+--java.net.URLConnection
|
+--java.net.HttpURLConnection
|
+--com.ibm.speech.net.www.protocol.https.HttpURLConnection

```

public class **HttpURLConnection**  
 extends java.net.HttpURLConnection

The class `HttpURLConnection` represents a communications link between the application and a URL. Instances of this class can be used both to read from and to write to the resource referenced by the URL. In general, creating a connection to a URL is a multistep process:

<code>openConnection()</code>	<code>connect()</code>
Manipulate parameters that affect the connection to the remote resource.	Interact with the resource; query header fields and contents.

time →

1. The connection object is created by invoking the `openConnection` method on a URL.
2. The setup parameters and general request properties are manipulated.
3. The actual connection to the remote object is made, using the `connect` method.
4. The remote object becomes available. The header fields and the contents of the remote object can be accessed.

The setup parameters are modified using the following methods:

- `setAllowUserInteraction`
- `setDoInput`
- `setDoOutput`
- `setIfModifiedSince`
- `setUseCaches`

and the general request properties are modified using the method:

- `setRequestProperty`

Default values for the `AllowUserInteraction` and `UseCaches` parameters can be set using the methods `setDefaultAllowUserInteraction` and `setDefaultUseCaches`. Default values for general request properties can be set using the `setDefaultRequestProperty` method.

Each of the above `set` methods has a corresponding `get` method to retrieve the value of the parameter or general request property. The specific parameters and general request properties that are applicable are protocol specific.

The following methods are used to access the header fields and the contents after the connection is made to the remote object:

- `getContent`
- `getHeaderField`
- `getInputStream`
- `getOutputStream`

Certain header fields are accessed frequently. The methods:

- `getContentEncoding`
- `getContentLength`
- `getContentType`
- `getDate`
- `getExpiration`
- `getLastModified`

provide convenient access to these fields. The `getContentType` method is used by the `getContent` method to determine the type of the remote object; subclasses may find it convenient to override the `getContentType` method.

In the common case, all of the pre-connection parameters and general request properties can be ignored: the pre-connection parameters and request properties default to sensible values. For most clients of this interface, there are only two interesting methods: `getInputStream` and `getObject`, which are mirrored in the `URL` class by convenience methods.

`HttpURLConnection` is a `URLConnection` with support for HTTPS-specific features. See [the spec](#) for details.

## Field Summary

static int	<u><code>HTTP_ACCEPTED</code></u>
static int	<u><code>HTTP_BAD_GATEWAY</code></u>
static int	<u><code>HTTP_BAD_METHOD</code></u>
static int	<u><code>HTTP_BAD_REQUEST</code></u> 4XX: client error
static int	<u><code>HTTP_CLIENT_TIMEOUT</code></u>
static int	<u><code>HTTP_CONFLICT</code></u>
static int	<u><code>HTTP_CREATED</code></u>

static int	<u>HTTP_ENTITY_TOO_LARGE</u>
static int	<u>HTTP_FORBIDDEN</u>
static int	<u>HTTP_GATEWAY_TIMEOUT</u>
static int	<u>HTTP_GONE</u>
static int	<u>HTTP_INTERNAL_ERROR</u>
static int	<u>HTTP_LENGTH_REQUIRED</u>
static int	<u>HTTP_MOVED_PERM</u>
static int	<u>HTTP_MOVED_TEMP</u>
static int	<u>HTTP_MULT_CHOICE</u> 3XX: relocation/redirect
static int	<u>HTTP_NO_CONTENT</u>
static int	<u>HTTP_NOT_ACCEPTABLE</u>
static int	<u>HTTP_NOT_AUTHORITY</u>
static int	<u>HTTP_NOT_FOUND</u>
static int	<u>HTTP_NOT_MODIFIED</u>
static int	<u>HTTP_OK</u> 2XX: generally "OK"
static int	<u>HTTP_PARTIAL</u>
static int	<u>HTTP_PAYMENT_REQUIRED</u>
static int	<u>HTTP_PRECON_FAILED</u>
static int	<u>HTTP_PROXY_AUTH</u>
static int	<u>HTTP_REQ_TOO_LONG</u>
static int	<u>HTTP_RESET</u>
static int	<u>HTTP_SEE_OTHER</u>

static int	<u>HTTP_SERVER_ERROR</u> 5XX: server error
static int	<u>HTTP_UNAUTHORIZED</u>
static int	<u>HTTP_UNAVAILABLE</u>
static int	<u>HTTP_UNSUPPORTED_TYPE</u>
static int	<u>HTTP_USE_PROXY</u>
static int	<u>HTTP_VERSION</u>

### Fields inherited from class java.net.HttpURLConnection

HTTP\_ACCEPTED, HTTP\_BAD\_GATEWAY, HTTP\_BAD\_METHOD, HTTP\_BAD\_REQUEST, HTTP\_CLIENT\_TIMEOUT, HTTP\_CONFLICT, HTTP\_CREATED, HTTP\_ENTITY\_TOO\_LARGE, HTTP\_FORBIDDEN, HTTP\_GATEWAY\_TIMEOUT, HTTP\_GONE, HTTP\_INTERNAL\_ERROR, HTTP\_LENGTH\_REQUIRED, HTTP\_MOVED\_PERM, HTTP\_MOVED\_TEMP, HTTP\_MULT\_CHOICE, HTTP\_NO\_CONTENT, HTTP\_NOT\_ACCEPTABLE, HTTP\_NOT\_AUTHENTICATED, HTTP\_NOT\_FOUND, HTTP\_NOT\_MODIFIED, HTTP\_OK, HTTP\_PARTIAL, HTTP\_PAYMENT\_REQUIRED, HTTP\_PRECON\_FAILED, HTTP\_PROXY\_AUTH, HTTP\_REQ\_TOO\_LONG, HTTP\_RESET, HTTP\_SEE\_OTHER, HTTP\_SERVER\_ERROR, HTTP\_UNAUTHORIZED, HTTP\_UNAVAILABLE, HTTP\_UNSUPPORTED\_TYPE, HTTP\_USE\_PROXY, HTTP\_VERSION, method, responseCode, responseMessage

### Fields inherited from class java.net.URLConnection

allowUserInteraction, connected, doInput, doOutput, ifModifiedSince, url, useCaches

## Constructor Summary

**HttpURLConnection**(java.net.URL u)

Creates a new HttpURLConnection instance to the object referenced by the URL argument with the default debug flag.

**HttpURLConnection**(java.net.URL u, boolean dbg)

Creates a new HttpURLConnection instance to the object referenced by the URL argument with a specified debug flag.

## Method Summary

void	<b><u>connect</u></b> () Opens a communications link to the resource referenced by this URL, if such a connection has not already been established.
void	<b><u>disconnect</u></b> () Close the connection to the server.
boolean	<b><u>getAllowUserInteraction</u></b> () Returns the value of the allowUserInteraction field for this object.

java.lang.String	<u>getContentEncoding()</u> Returns the value of the content-encoding header field.
int	<u>getContentLength()</u> Returns the value of the content-length header field.
java.lang.String	<u>getContentType()</u> Returns the value of the content-type header field.
long	<u>getDate()</u> Returns the value of the date header field.
static boolean	<u>getDefaultAllowUserInteraction()</u> Returns the default value of the allowUserInteraction field.
static java.lang.String	<u>getDefaultRequestProperty</u> (java.lang.String key) Returns the value of the default request property.
boolean	<u>getDefaultUseCaches()</u> Returns the default value of a URLConnection's useCaches flag.
boolean	<u>getDoInput()</u> Returns the value of this URLConnection's doInput flag.
boolean	<u>getDoOutput()</u> Returns the value of this URLConnection's doOutput flag.
java.io.InputStream	<u>getErrorStream()</u> Returns the error stream if the connection failed but the server sent useful data nonetheless.
long	<u>getExpiration()</u> Returns the value of the expires header field.
static java.net.FileNameMap	<u>getFileNameMap()</u> Returns the FileNameMap.
static boolean	<u>getFollowRedirects()</u>
java.lang.String	<u>getHeaderField</u> (int n) Returns the value for the n <sup>th</sup> header field.
java.lang.String	<u>getHeaderField</u> (java.lang.String name) Returns the name of the specified header field.
long	<u>getHeaderFieldDate</u> (java.lang.String name, long Default) Returns the value of the named field parsed as date.
int	<u>getHeaderFieldInt</u> (java.lang.String name, int Default) Returns the value of the named field parsed as a number.
java.lang.String	<u>getHeaderFieldKey</u> (int n) Returns the key for the n <sup>th</sup> header field.
java.lang.String	<u>getHTTPHeader()</u> Returns the entire of the HTTP header received from server request.
long	<u>getIfModifiedSince()</u> Returns the value of this object's ifModifiedSince field.
java.io.InputStream	<u>getInputStream()</u> Returns an input stream that reads from this open connection.

long	<u><b>getLastModified()</b></u> Returns the value of the last-modified header field.
java.io.OutputStream	<u><b>getOutputStream()</b></u> Returns an output stream that writes to this connection.
java.security.Permission	<u><b>getPermission()</b></u> Returns a permission object representing the permission necessary to make the connection represented by this object.
java.lang.String	<u><b>getRequestMethod()</b></u> Get the request method.
java.lang.String	<u><b>getRequestProperty()</b></u> (java.lang.String key) Returns the value of the named general request property for this connection.
int	<u><b>getResponseCode()</b></u> Gets HTTP response status.
java.lang.String	<u><b>getResponseMessage()</b></u> Gets the HTTP response message, if any, returned along with the response code from a server.
java.net.URL	<u><b>getURL()</b></u> Returns the value of this URLConnection's URL field.
boolean	<u><b>getUseCaches()</b></u> Returns the value of this URLConnection's useCaches field.
protected static java.lang.String	<u><b>guessContentTypeFromName()</b></u> (java.lang.String fname) Tries to determine the content type of an object, based on the specified "file" component of a URL.
static java.lang.String	<u><b>guessContentTypeFromStream()</b></u> (java.io.InputStream is) Tries to determine the type of an input stream based on the characters at the beginning of the input stream.
void	<u><b>setAllowUserInteraction()</b></u> (boolean allowuserinteraction) Set the value of the allowUserInteraction field of this URLConnection.
void	<u><b>setAsynConnections()</b></u> (boolean value) Sets the value of the asyncConnectionSet field for the HttpURLConnection object to the specified value.
static void	<u><b>setContentHandlerFactory()</b></u> (java.net.ContentHandlerFactory fac) Sets the ContentHandlerFactory of an application.
static void	<u><b>setDefaultAllowUserInteraction()</b></u> (boolean defaultallowuserinteraction) Sets the default value of the allowUserInteraction field for all future URLConnection objects to the specified value.
static void	<u><b>setDefaultRequestProperty()</b></u> (java.lang.String key, java.lang.String value) Sets the default value of a general request property.
void	<u><b>setDefaultUseCaches()</b></u> (boolean defaultusecaches) Sets the default value of the useCaches field to the specified value.

void	<b><u>setDoInput</u></b> (boolean doinput) Sets the value of the doInput field for this HttpURLConnection to the specified value.
void	<b><u>setDoOutput</u></b> (boolean dooutput) Sets the value of the doOutput field for this HttpURLConnection to the specified value.
void	<b><u>setEnabledCipherSuites</u></b> (java.lang.String cipherSuites) Sets the value of the enabledCipherSuites field for the HttpURLConnection object to the specified value.
void	<b><u>setEnabledCompressionMethods</u></b> (java.lang.String methods) Sets the value of the enabledCompressionMethods field for the HttpURLConnection object to the specified value.
static void	<b><u>setFileNameMap</u></b> (java.net.FileNameMap map) Sets the FileNameMap.
static void	<b><u>setFollowRedirects</u></b> (boolean set) Sets whether HTTP redirects (requests with response code 3xx) should be automatically followed by this class.
void	<b><u>setIfModifiedSince</u></b> (long ifmodifiedsince) Sets the value of the ifModifiedSince field of this HttpURLConnection to the specified value.
void	<b><u>setKeyRingDatabase</u></b> (java.lang.String name) Sets the value of the keyRingDatabase field for the HttpURLConnection object to the specified value.
void	<b><u>setRequestMethod</u></b> (java.lang.String method) Set the method for the URL request, one of: GET POST HEAD OPTIONS PUT DELETE TRACE are legal, subject to protocol restrictions.
void	<b><u>setRequestProperty</u></b> (java.lang.String key, java.lang.String value) Sets the general request property.
void	<b><u>setTimeout</u></b> (int seconds) Sets the value of the timeout field for the HttpURLConnection object to the specified value.
void	<b><u>setUseCaches</u></b> (boolean usecaches) Sets the value of the useCaches field of this URLConnection to the specified value.
java.lang.String	<b><u>toString</u></b> () Returns a String representation of this URL connection.
boolean	<b><u>usingProxy</u></b> () Indicates if the connection is going through a proxy.

**Methods inherited from class java.net.URLConnection**

getContent

**Methods inherited from class java.lang.Object**

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait



## Field Detail

### HTTP\_OK

```
public static final int HTTP_OK
```

2XX: generally "OK"

### HTTP\_CREATED

```
public static final int HTTP_CREATED
```

### HTTP\_ACCEPTED

```
public static final int HTTP_ACCEPTED
```

### HTTP\_NOT\_AUTHORITATIVE

```
public static final int HTTP_NOT_AUTHORITATIVE
```

### HTTP\_NO\_CONTENT

```
public static final int HTTP_NO_CONTENT
```

### HTTP\_RESET

```
public static final int HTTP_RESET
```

### HTTP\_PARTIAL

```
public static final int HTTP_PARTIAL
```

### HTTP\_MULT\_CHOICE

```
public static final int HTTP_MULT_CHOICE
```

3XX: relocation/redirect

**HTTP\_MOVED\_PERM**

```
public static final int HTTP_MOVED_PERM
```

---

**HTTP\_MOVED\_TEMP**

```
public static final int HTTP_MOVED_TEMP
```

---

**HTTP\_SEE\_OTHER**

```
public static final int HTTP_SEE_OTHER
```

---

**HTTP\_NOT\_MODIFIED**

```
public static final int HTTP_NOT_MODIFIED
```

---

**HTTP\_USE\_PROXY**

```
public static final int HTTP_USE_PROXY
```

---

**HTTP\_BAD\_REQUEST**

```
public static final int HTTP_BAD_REQUEST
```

4XX: client error

---

**HTTP\_UNAUTHORIZED**

```
public static final int HTTP_UNAUTHORIZED
```

---

**HTTP\_PAYMENT\_REQUIRED**

```
public static final int HTTP_PAYMENT_REQUIRED
```

---

**HTTP\_FORBIDDEN**

```
public static final int HTTP_FORBIDDEN
```

---

## **HTTP\_NOT\_FOUND**

```
public static final int HTTP_NOT_FOUND
```

---

## **HTTP\_BAD\_METHOD**

```
public static final int HTTP_BAD_METHOD
```

---

## **HTTP\_NOT\_ACCEPTABLE**

```
public static final int HTTP_NOT_ACCEPTABLE
```

---

## **HTTP\_PROXY\_AUTH**

```
public static final int HTTP_PROXY_AUTH
```

---

## **HTTP\_CLIENT\_TIMEOUT**

```
public static final int HTTP_CLIENT_TIMEOUT
```

---

## **HTTP\_CONFLICT**

```
public static final int HTTP_CONFLICT
```

---

## **HTTP\_GONE**

```
public static final int HTTP_GONE
```

---

## **HTTP\_LENGTH\_REQUIRED**

```
public static final int HTTP_LENGTH_REQUIRED
```

---

## **HTTP\_PRECON\_FAILED**

```
public static final int HTTP_PRECON_FAILED
```

---

## **HTTP\_ENTITY\_TOO\_LARGE**

```
public static final int HTTP_ENTITY_TOO_LARGE
```

---

## HTTP\_REQ\_TOO\_LONG

```
public static final int HTTP_REQ_TOO_LONG
```

---

## HTTP\_UNSUPPORTED\_TYPE

```
public static final int HTTP_UNSUPPORTED_TYPE
```

---

## HTTP\_SERVER\_ERROR

```
public static final int HTTP_SERVER_ERROR
```

5XX: server error

---

## HTTP\_INTERNAL\_ERROR

```
public static final int HTTP_INTERNAL_ERROR
```

---

## HTTP\_BAD\_GATEWAY

```
public static final int HTTP_BAD_GATEWAY
```

---

## HTTP\_UNAVAILABLE

```
public static final int HTTP_UNAVAILABLE
```

---

## HTTP\_GATEWAY\_TIMEOUT

```
public static final int HTTP_GATEWAY_TIMEOUT
```

---

## HTTP\_VERSION

```
public static final int HTTP_VERSION
```

## Constructor Detail

## HttpsURLConnection

```
public HttpURLConnection(java.net.URL u)
```

Creates a new `HttpsURLConnection` instance to the object referenced by the `URL` argument with the default debug flag.

**Parameters:**

`u` - the `URL` that this connects to.

---

## HttpsURLConnection

```
public HttpURLConnection(java.net.URL u,  
                          boolean dbg)
```

Creates a new `HttpsURLConnection` instance to the object referenced by the `URL` argument with a specified debug flag.

**Parameters:**

`u` - the `URL` that this connects to.

`dbg` - `dbg` the boolean value of turning the debug option ON/OFF.

## Method Detail

### getInputStream

```
public java.io.InputStream getInputStream()  
                           throws java.io.IOException
```

Returns an input stream that reads from this open connection.

**Overrides:**

`getInputStream` in class `java.net.URLConnection`

**Returns:**

an input stream that reads from this open connection.

**Throws:**

`java.io.IOException` - if an I/O error occurs while creating the input stream.

`UnknownServiceException` - if the protocol does not support input.

---

### getOutputStream

```
public java.io.OutputStream getOutputStream()  
                           throws java.io.IOException
```

Returns an output stream that writes to this connection.

**Overrides:**

`getOutputStream` in class `java.net.URLConnection`

**Returns:**

an output stream that writes to this connection.

**Throws:**

`java.io.IOException` - if an I/O error occurs while creating the output stream.

`UnknownServiceException` - if the protocol does not support output.

---

## disconnect

```
public void disconnect()
```

Close the connection to the server.

**Overrides:**

disconnect in class java.net.HttpURLConnection

---

## usingProxy

```
public boolean usingProxy()
```

Indicates if the connection is going through a proxy.

**Overrides:**

usingProxy in class java.net.HttpURLConnection

---

## connect

```
public void connect()  
    throws java.io.IOException
```

Opens a communications link to the resource referenced by this URL, if such a connection has not already been established.

If the connect method is called when the connection has already been opened (indicated by the connected field having the value true), the call is ignored.

HttpURLConnection objects go through two phases: first they are created, then they are connected. After being created, and before being connected, various options can be specified (e.g., doInput and UseCaches). After connecting, it is an error to try to set them. Operations that depend on being connected, like getContentLength, will implicitly perform the connection, if necessary.

**Overrides:**

connect in class java.net.URLConnection

**Throws:**

java.io.IOException - if an I/O error occurs while opening the connection.

---

## setRequestMethod

```
public void setRequestMethod(java.lang.String method)  
    throws java.net.ProtocolException
```

Set the method for the URL request, one of:

- GET
- POST
- HEAD

- OPTIONS
- PUT
- DELETE
- TRACE

are legal, subject to protocol restrictions. The default method is GET.

**Overrides:**

setRequestMethod in class java.net.HttpURLConnection

**Throws:**

java.net.ProtocolException - if the method cannot be reset or if the requested method isn't valid for HTTP.

## setRequestProperty

```
public void setRequestProperty(java.lang.String key,
                               java.lang.String value)
```

Sets the general request property.

**Overrides:**

setRequestProperty in class java.net.URLConnection

**Parameters:**

key - the keyword by which the request is known (e.g., "accept").  
value - the value associated with it.

## getHeaderField

```
public java.lang.String getHeaderField(java.lang.String name)
```

Returns the name of the specified header field.

**Overrides:**

getHeaderField in class java.net.URLConnection

**Parameters:**

name - the name of a header field.

**Returns:**

the value of the named header field, or null if there is no such field in the header.

## getHTTPHeader

```
public java.lang.String getHTTPHeader()
```

Returns the entire of the HTTP header received from server request.

**Returns:**

the value of the httpHeader field for this object.

## getAllowUserInteraction

```
public boolean getAllowUserInteraction()
```

Returns the value of the allowUserInteraction field for this object.

**Overrides:**

getAllowUserInteraction in class java.net.URLConnection

**Returns:**

the value of the allowUserInteraction field for this object.

---

**getContentEncoding**

```
public java.lang.String getContentEncoding()
```

Returns the value of the content-encoding header field.

**Overrides:**

getContentEncoding in class java.net.URLConnection

**Returns:**

the content encoding of the resource that the URL references, or null if not known.

---

**getContentLength**

```
public int getContentLength()
```

Returns the value of the content-length header field.

**Overrides:**

getContentLength in class java.net.URLConnection

**Returns:**

the content length of the resource that this connection's URL references, or -1 if the content length is not known.

---

**getContentType**

```
public java.lang.String getContentType()
```

Returns the value of the content-type header field.

**Overrides:**

getContentType in class java.net.URLConnection

**Returns:**

the content type of the resource that the URL references, or null if not known.

---

**getDate**

```
public long getDate()
```

Returns the value of the date header field.

**Overrides:**

getDate in class java.net.URLConnection

**Returns:**

the sending date of the resource that the URL references, or 0 if not known. The value returned is the number of milliseconds since January 1, 1970 GMT.



## getDefaultAllowUserInteraction

```
public static boolean getDefaultAllowUserInteraction()
```

Returns the default value of the `allowUserInteraction` field.

This default is "sticky", being a part of the static state of all `URLConnection`s. This flag applies to the next, and all following `URLConnection`s that are created.

**Returns:**

the default value of the `allowUserInteraction` field.

---

## getDefaultRequestProperty

```
public static java.lang.String getDefaultRequestProperty(java.lang.String key)
```

Returns the value of the default request property. Default request properties are set for every connection.

**Returns:**

the value of the default request property for the specified key.

---

## getDefaultUseCaches

```
public boolean getDefaultUseCaches()
```

Returns the default value of a `URLConnection`'s `useCaches` flag.

This default is "sticky", being a part of the static state of all `URLConnection`s. This flag applies to the next, and all following `URLConnection`s that are created.

**Overrides:**

`getDefaultUseCaches` in class `java.net.URLConnection`

**Returns:**

the default value of a `URLConnection`'s `useCaches` flag.

---

## getDoInput

```
public boolean getDoInput()
```

Returns the value of this `URLConnection`'s `doInput` flag.

**Overrides:**

`getDoInput` in class `java.net.URLConnection`

**Returns:**

the value of this `URLConnection`'s `doInput` flag.

---

## getDoOutput

```
public boolean getDoOutput()
```

Returns the value of this `URLConnection`'s `doOutput` flag.

**Overrides:**

`getDoOutput` in class `java.net.URLConnection`

**Returns:**

the value of this `URLConnection`'s `doOutput` flag.

---

## **getExpiration**

```
public long getExpiration()
```

Returns the value of the `expires` header field.

**Overrides:**

`getExpiration` in class `java.net.URLConnection`

**Returns:**

the expiration date of the resource that this URL references, or 0 if not known. The value is the number of milliseconds since January 1, 1970 GMT.

---

## **getFileNameMap**

```
public static java.net.FileNameMap getFileNameMap()
```

Returns the `FileNameMap`.

**Since:**

JDK1.2

---

## **getHeaderField**

```
public java.lang.String getHeaderField(int n)
```

Returns the value for the  $n^{\text{th}}$  header field. It returns `null` if there are fewer than  $n$  fields.

This method can be used in conjunction with the `getHeaderFieldKey` method to iterate through all the headers in the message.

**Overrides:**

`getHeaderField` in class `java.net.URLConnection`

**Parameters:**

$n$  - an index.

**Returns:**

the value of the  $n^{\text{th}}$  header field.

---

## **getHeaderFieldDate**

```
public long getHeaderFieldDate(java.lang.String name,
```

long Default)

Returns the value of the named field parsed as date. The result is the number of milliseconds since January 1, 1970 GMT represented by the named field.

This form of `getHeaderField` exists because some connection types (e.g., `http-ng`) have pre-parsed headers. Classes for that connection type can override this method and short-circuit the parsing.

**Overrides:**

`getHeaderFieldDate` in class `java.net.URLConnection`

**Parameters:**

`name` - the name of the header field.

`Default` - a default value.

**Returns:**

the value of the field, parsed as a date. The value of the `Default` argument is returned if the field is missing or malformed.

## **getHeaderFieldInt**

```
public int getHeaderFieldInt(java.lang.String name,
                             int Default)
```

Returns the value of the named field parsed as a number.

This form of `getHeaderField` exists because some connection types (e.g., `http-ng`) have pre-parsed headers. Classes for that connection type can override this method and short-circuit the parsing.

**Overrides:**

`getHeaderFieldInt` in class `java.net.URLConnection`

**Parameters:**

`name` - the name of the header field.

`Default` - the default value.

**Returns:**

the value of the named field, parsed as an integer. The `Default` value is returned if the field is missing or malformed.

## **getHeaderFieldKey**

```
public java.lang.String getHeaderFieldKey(int n)
```

Returns the key for the  $n^{\text{th}}$  header field.

**Overrides:**

`getHeaderFieldKey` in class `java.net.URLConnection`

**Parameters:**

`n` - an index.

**Returns:**

the key for the  $n^{\text{th}}$  header field, or `null` if there are fewer than `n` fields.

## getIfModifiedSince

```
public long getIfModifiedSince()
```

Returns the value of this object's `ifModifiedSince` field.

**Overrides:**

`getIfModifiedSince` in class `java.net.URLConnection`

**Returns:**

the value of this object's `ifModifiedSince` field.

---

## getLastModified

```
public long getLastModified()
```

Returns the value of the `last-modified` header field. The result is the number of milliseconds since January 1, 1970 GMT.

**Overrides:**

`getLastModified` in class `java.net.URLConnection`

**Returns:**

the date the resource referenced by this `URLConnection` was last modified, or 0 if not known.

---

## getPermission

```
public java.security.Permission getPermission()  
    throws java.io.IOException
```

Returns a permission object representing the permission necessary to make the connection represented by this object. This method returns null if no permission is required to make the connection. By default, this method returns `java.security.AllPermission`. Subclasses should override this method and return the permission that best represents the permission required to make a connection to the URL. For example, a `URLConnection` representing a file: URL would return a `java.io.FilePermission` object.

The permission returned may dependent upon the state of the connection. For example, the permission before connecting may be different from that after connecting. For example, an HTTP sever, say `foo.com`, may redirect the connection to a different host, say `bar.com`. Before connecting the permission returned by the connection will represent the permission needed to connect to `foo.com`, while the permission returned after connecting will be to `bar.com`.

Permissions are generally used for two purposes: to protect caches of objects obtained through `URLConnections`, and to check the right of a recipient to learn about a particular URL. In the first case, the permission should be obtained *after* the object has been obtained. For example, in an HTTP connection, this will represent the permission to connect to the host from which the data was ultimately fetched. In the second case, the permission should be obtained and tested *before* connecting.

**Overrides:**

`getPermission` in class `java.net.HttpURLConnection`

**Returns:**

the permission object representing the permission necessary to make the connection represented by this `URLConnection`.

**Throws:**

`java.io.IOException` - if the computation of the permission requires network or file I/O and an exception occurs while computing it.

---

## **getRequestProperty**

```
public java.lang.String getRequestProperty(java.lang.String key)
```

Returns the value of the named general request property for this connection.

**Overrides:**

`getRequestProperty` in class `java.net.URLConnection`

**Returns:**

the value of the named general request property for this connection.

---

## **getURL**

```
public java.net.URL getURL()
```

Returns the value of this `URLConnection`'s `URL` field.

**Overrides:**

`getURL` in class `java.net.URLConnection`

**Returns:**

the value of this `URLConnection`'s `URL` field.

---

## **getUseCaches**

```
public boolean getUseCaches()
```

Returns the value of this `URLConnection`'s `useCaches` field.

**Overrides:**

`getUseCaches` in class `java.net.URLConnection`

**Returns:**

the value of this `URLConnection`'s `useCaches` field.

---

## **guessContentTypeFromName**

```
protected static java.lang.String guessContentTypeFromName(java.lang.String fname)
```

Tries to determine the content type of an object, based on the specified "file" component of a URL. This is a convenience method that can be used by subclasses that override the `getContentType` method.

**Parameters:**

`fname` - a filename.

**Returns:**

a guess as to what the content type of the object is, based upon its file name.

---

## guessContentTypeFromStream

```
public static java.lang.String guessContentTypeFromStream(java.io.InputStream is)
                                throws java.io.IOException
```

Tries to determine the type of an input stream based on the characters at the beginning of the input stream. This method can be used by subclasses that override the `getContentType` method.

Ideally, this routine would not be needed. But many http servers return the incorrect content type; in addition, there are many nonstandard extensions. Direct inspection of the bytes to determine the content type is often more accurate than believing the content type claimed by the http server.

**Parameters:**

`is` - an input stream that supports marks.

**Returns:**

a guess at the content type, or null if none can be determined.

**Throws:**

`java.io.IOException` - if an I/O error occurs while reading the input stream.

---

## setAllowUserInteraction

```
public void setAllowUserInteraction(boolean allowuserinteraction)
```

Set the value of the `allowUserInteraction` field of this `URLConnection`.

**Overrides:**

`setAllowUserInteraction` in class `java.net.URLConnection`

**Parameters:**

`allowuserinteraction` - the new value.

---

## setContentHandlerFactory

```
public static void setContentHandlerFactory(java.net.ContentHandlerFactory fac)
```

Sets the `ContentHandlerFactory` of an application. It can be called at most once by an application.

The `ContentHandlerFactory` instance is used to construct a content handler from a content type

If there is a security manager, this method first calls the security manager's `checkSetFactory` method to ensure the operation is allowed. This could result in a `SecurityException`.

**Parameters:**

`fac` - the desired factory.

**Throws:**

`java.lang.Error` - if the factory has already been defined.

`java.lang.SecurityException` - if a security manager exists and its `checkSetFactory` method doesn't allow the operation.

---

## setDefaultAllowUserInteraction

```
public static void setDefaultAllowUserInteraction(boolean defaultallowuserinteracti
```

Sets the default value of the `allowUserInteraction` field for all future `URLConnection` objects to the specified value.

**Parameters:**

`defaultallowuserinteraction` - the new value.

---

## setDefaultRequestProperty

```
public static void setDefaultRequestProperty(java.lang.String key,  
                                             java.lang.String value)
```

Sets the default value of a general request property. When a `URLConnection` is created, it is initialized with these properties.

**Parameters:**

`key` - the keyword by which the request is known (e.g., "accept").  
`value` - the value associated with the key.

---

## setDefaultUseCaches

```
public void setDefaultUseCaches(boolean defaultusecaches)
```

Sets the default value of the `useCaches` field to the specified value.

**Overrides:**

`setDefaultUseCaches` in class `java.net.URLConnection`

**Parameters:**

`defaultusecaches` - the new value.

---

## setDoInput

```
public void setDoInput(boolean doinput)
```

Sets the value of the `doInput` field for this `URLConnection` to the specified value.

A URL connection can be used for input and/or output. Set the `DoInput` flag to true if you intend to use the URL connection for input, false if not. The default is true unless `DoOutput` is explicitly set to true, in which case `DoInput` defaults to false.

**Overrides:**

`setDoInput` in class `java.net.URLConnection`

**Parameters:**

`value` - the new value.

---

## setDoOutput

```
public void setDoOutput(boolean dooutput)
```

Sets the value of the `doOutput` field for this `URLConnection` to the specified value.

A URL connection can be used for input and/or output. Set the `DoOutput` flag to true if you intend to use the URL connection for output, false if not. The default is false.

**Overrides:**

`setDoOutput` in class `java.net.URLConnection`

**Parameters:**

`value` - the new value.

---

## setFileNameMap

```
public static void setFileNameMap(java.net.FileNameMap map)
```

Sets the `FileNameMap`.

If there is a security manager, this method first calls the security manager's `checkSetFactory` method to ensure the operation is allowed. This could result in a `SecurityException`.

**Parameters:**

`map` - the `FileNameMap` to be set

**Throws:**

`java.lang.SecurityException` - if a security manager exists and its `checkSetFactory` method doesn't allow the operation.

---

## setIfModifiedSince

```
public void setIfModifiedSince(long ifmodifiedsince)
```

Sets the value of the `ifModifiedSince` field of this `URLConnection` to the specified value.

**Overrides:**

`setIfModifiedSince` in class `java.net.URLConnection`

**Parameters:**

`value` - the new value.

---

## setUseCaches

```
public void setUseCaches(boolean usecaches)
```

Sets the value of the `useCaches` field of this `URLConnection` to the specified value.

Some protocols do caching of documents. Occasionally, it is important to be able to "tunnel through" and ignore the caches (e.g., the "reload" button in a browser). If the `UseCaches` flag on a connection is true, the connection is allowed to use whatever caches it can. If false, caches



are to be ignored. The default value comes from `DefaultUseCaches`, which defaults to true.

**Overrides:**

`setUseCaches` in class `java.net.URLConnection`

---

**toString**

```
public java.lang.String toString()
```

Returns a `String` representation of this URL connection.

**Overrides:**

`toString` in class `java.net.URLConnection`

**Returns:**

a string representation of this `URLConnection`.

---

**getErrorStream**

```
public java.io.InputStream getErrorStream()
```

Returns the error stream if the connection failed but the server sent useful data nonetheless. The typical example is when an HTTP server responds with a 404, which will cause a `FileNotFoundException` to be thrown in connect, but the server sent an HTML help page with suggestions as to what to do.

This method will not cause a connection to be initiated. If there the connection was not connected, or if the server did not have an error while connecting or if the server did have an error but there no error data was sent, this method will return null. This is the default.

**Overrides:**

`getErrorStream` in class `java.net.HttpURLConnection`

**Returns:**

an error stream if any, null if there have been no errors, the connection is not connected or the server sent no useful data.

---

**getFollowRedirects**

```
public static boolean getFollowRedirects()
```

---

**getRequestMethod**

```
public java.lang.String getRequestMethod()
```

Get the request method.

**Overrides:**

`getRequestMethod` in class `java.net.HttpURLConnection`

---

## getResponseCode

```
public int getResponseCode()
    throws java.io.IOException
```

Gets HTTP response status. From responses like:

```
HTTP/1.0 200 OK
HTTP/1.0 401 Unauthorized
```

Extracts the ints 200 and 401 respectively. Returns -1 if none can be discerned from the response (i.e., the response is not valid HTTP).

**Overrides:**

getResponseCode in class java.net.HttpURLConnection

**Throws:**

java.io.IOException - if an error occurred connecting to the server.

## getResponseMessage

```
public java.lang.String getResponseMessage()
    throws java.io.IOException
```

Gets the HTTP response message, if any, returned along with the response code from a server. From responses like:

```
HTTP/1.0 200 OK
HTTP/1.0 404 Not Found
```

Extracts the Strings "OK" and "Not Found" respectively. Returns null if none could be discerned from the responses (the result was not valid HTTP).

**Overrides:**

getResponseMessage in class java.net.HttpURLConnection

**Throws:**

java.io.IOException - if an error occurred connecting to the server.

## setFollowRedirects

```
public static void setFollowRedirects(boolean set)
```

Sets whether HTTP redirects (requests with response code 3xx) should be automatically followed by this class. True by default. Applets cannot change this variable.

If there is a security manager, this method first calls the security manager's checkSetFactory method to ensure the operation is allowed. This could result in a SecurityException.

**Throws:**

java.lang.SecurityException - if a security manager exists and its checkSetFactory method doesn't allow the operation.

---

## setKeyRingDatabase

```
public void setKeyRingDatabase(java.lang.String name)
```

Sets the value of the `keyRingDatabase` field for the `HttpURLConnection` object to the specified value. SSL specific API extension.

**Parameters:**

name - the new value.

---

## setTimeout

```
public void setTimeout(int seconds)
```

Sets the value of the `timeout` field for the `HttpURLConnection` object to the specified value. SSL specific API extension.

**Parameters:**

seconds - the new value.

---

## setAsynConnections

```
public void setAsynConnections(boolean value)
```

Sets the value of the `asyncConnectionSet` field for the `HttpURLConnection` object to the specified value. SSL specific API extension.

**Parameters:**

value - the new value.

---

## setEnabledCompressionMethods

```
public void setEnabledCompressionMethods(java.lang.String methods)
```

Sets the value of the `enabledCompressionMethods` field for the `HttpURLConnection` object to the specified value. SSL specific API extension.

**Parameters:**

methods - the new value.

---

## setEnabledCipherSuites

```
public void setEnabledCipherSuites(java.lang.String cipherSuites)
```

Sets the value of the `enabledCipherSuites` field for the `HttpURLConnection` object to the specified value. SSL specific API extension.

**Parameters:**

cipherSuites - the new value.

---

**PREV CLASS**   **NEXT CLASS**

<u>FRAMES</u>	<u>NO FRAMES</u>
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	11
12	12
13	13
14	14
15	15
16	16
17	17
18	18
19	19
20	20
21	21
22	22
23	23
24	24
25	25
26	26
27	27
28	28
29	29
30	30
31	31
32	32
33	33
34	34
35	35
36	36
37	37
38	38
39	39
40	40
41	41
42	42
43	43
44	44
45	45
46	46
47	47
48	48
49	49
50	50
51	51
52	52
53	53
54	54
55	55
56	56
57	57
58	58
59	59
60	60
61	61
62	62
63	63
64	64
65	65
66	66
67	67
68	68
69	69
70	70
71	71
72	72
73	73
74	74
75	75
76	76
77	77
78	78
79	79
80	80
81	81
82	82
83	83
84	84
85	85
86	86
87	87
88	88
89	89
90	90
91	91
92	92
93	93
94	94
95	95
96	96
97	97
98	98
99	99
100	100

DETAIL: FIELD | CONSTR | METHOD